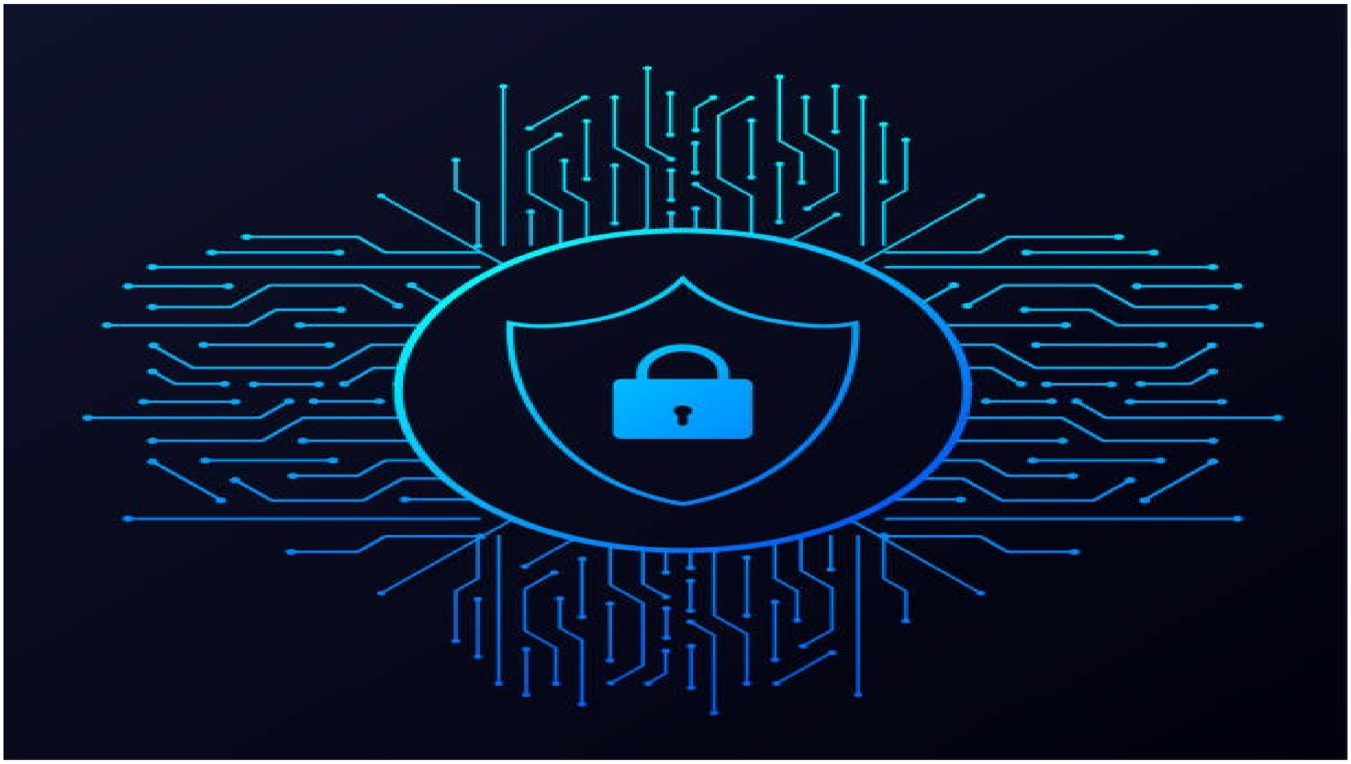


Coffee Corp
Security Assessment Findings Report

Business Confidential

Date: June 28th, 2024
Project: 897-19
Version 1.0

Table of Contents

Confidentiality Statement

This document is the exclusive property of Coffee Corp (CC) and Jigsaw64 Security (Jigsaw64S). This document contains proprietary and confidential information. Duplication, redistribution, or use, in whole or in part, in any form, requires consent of both CC and Jigsaw64S.

Jigsaw64S may share this document with auditors under non-disclosure agreements to demonstrate penetration test requirement compliance.

Disclaimer

A penetration test is considered a snapshot in time. The findings and recommendations reflect the information gathered during the assessment and not any changes or modifications made outside of that period.

Time-limited engagements do not allow for a full evaluation of all security controls. Jigsaw64S prioritized the assessment to identify the weakest security controls an attacker would exploit. Jigsaw64S recommends conducting similar assessments on an annual basis by internal or third-party assessors to ensure the continued success of the controls.

Contact Information

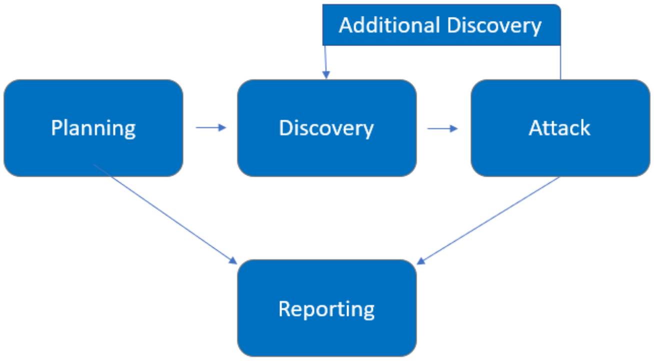
Name	Title	Contact Information
Coffee Corp		
John Smith	VP, Information Security (CISO)	Office: (555) 555-5555 Email: john.smith@demo.com
Jim Smith	IT Manager	Office: (555) 555-5555 Email: jim.smith@demo.com
Joe Smith	Network Engineer	Office: (555) 555-5555 Email: joe.smith@demo.com
Jigsaw64 Security		
CJ Oddo	Lead Penetration Tester	Office: (555) 555-5555 Email: coddo@Jigsaw64-sec.com
Bob Adams	Penetration Tester	Office: (555) 555-5555 Email: badams@Jigsaw64-sec.com
Rob Adams	Account Manager	Office: (555) 555-5555 Email: radams@Jigsaw64-sec.com

Assessment Overview

From May 20th, 2019 to May 29th, 2019, CC engaged Jigsaw64S to evaluate the security posture of its infrastructure compared to current industry best practices that included an Web Application penetration test. All testing performed is based on the NIST *SP 800-115 Technical Guide to Information Security Testing and Assessment, OWASP Testing Guide (v4), and customized testing frameworks*.

Phases of penetration testing activities include the following:

- Planning – Customer goals are gathered and rules of engagement obtained.
- Discovery – Perform scanning and enumeration to identify potential vulnerabilities, weak areas, and exploits.
- Attack – Confirm potential vulnerabilities through exploitation and perform additional discovery upon new access.
- Reporting – Document all found vulnerabilities and exploits, failed attempts, and company strengths and weaknesses.



Assessment Components

Web Application Penetration Test

An Web Application penetration test emulates the role of an attacker attempting to gain access to an internal network without internal resources or inside knowledge. A Jigsaw64S engineer attempts to gather sensitive information through open-source intelligence (OSINT), including employee information, historical breached passwords, and more that can be leveraged against Web Application systems to gain internal network access. The engineer also performs scanning and enumeration to identify potential vulnerabilities in hopes of exploitation.

Finding Severity Ratings

The following table defines levels of severity and corresponding CVSS score range that are used throughout the document to assess vulnerability and risk impact.

Severity	CVSS V3 Score Range	Definition
Critical	9.0-10.0	Exploitation is straightforward and usually results in system-level compromise. It is advised to form a plan of action and patch immediately.
High	7.0-8.9	Exploitation is more difficult but could cause elevated privileges and potentially a loss of data or downtime. It is advised to form a plan of action and patch as soon as possible.
Moderate	4.0-6.9	Vulnerabilities exist but are not exploitable or require extra steps such as social engineering. It is advised to form a plan of action and patch after high-priority issues have been resolved.
Low	0.1-3.9	Vulnerabilities are non-exploitable but would reduce an organization's attack surface. It is advised to form a plan of action and patch during the next maintenance window.
Informational	N/A	No vulnerability exists. Additional information is provided regarding items noticed during testing, strong controls, and additional documentation.

Scope

Assessment	Details
Web Application Penetration Test	
Container ID	39a87be399c
Web Application URL	http://localhost

- Full scope information provided in “**Coffee Corp-867-19 Full Findings.xslx”**

Scope Exclusions

Per client request, Jigsaw64S did not perform any Denial of Service attacks during testing.

Client Allowances

CC did not provide any allowances to assist the testing.

Executive Summary

Jigsaw64S evaluated CC's web application security posture through a network penetration test from June 25th, 2024, to June 25th, 2024. By leveraging a series of attacks, Jigsaw64S found critical-level vulnerabilities that allowed obtaining a shell on the container with ID 39a87be399c, which hosts the web application. It is highly recommended that CC address these vulnerabilities as soon as possible, as they are easily found through basic reconnaissance and exploitable without much effort.

Attack Summary

The following table describes how Jigsaw64S gained internal network access, step by step:

Step	Action	Recommendation
1	Discovered an SQL injection vulnerability in the web application, allowing data extraction	Implement input validation and prepared statements to mitigate SQL injection vulnerabilities.
2	Cracked admin password hashes, gaining credentials	Enforce a strong password policy (e.g, minimum 14 characters, use of special characters, no common words or phrases).
3	Used the cracked credentials to authenticate and upload a malicious reverse shell file.	Implement file upload restrictions and validate file types. Monitor for unauthorized file uploads and use antivirus solutions.
4	Executed the reverse shell, gaining remote access to the container.	Deploy security monitoring and intrusion detection systems to detect and respond to suspicious activities. Conduct regular security audits and penetration testing.

Security Strengths

Account Lockout Mechanism

One notable security strength was the implementation of an account lockout mechanism. Accounts were locked after five unsuccessful login attempts, effectively mitigating the risk of brute-force attacks on these accounts. This measure helps prevent unauthorized access through repeated login attempts and enhances the overall security of the system.

Security Weaknesses

SQL Injection Vulnerability

Jigsaw64S discovered SQL injection vulnerabilities in CC's web application, allowing unauthorized access to sensitive data. By exploiting these vulnerabilities, Jigsaw64S was able to extract password hashes from the database.

Weak Password Policy

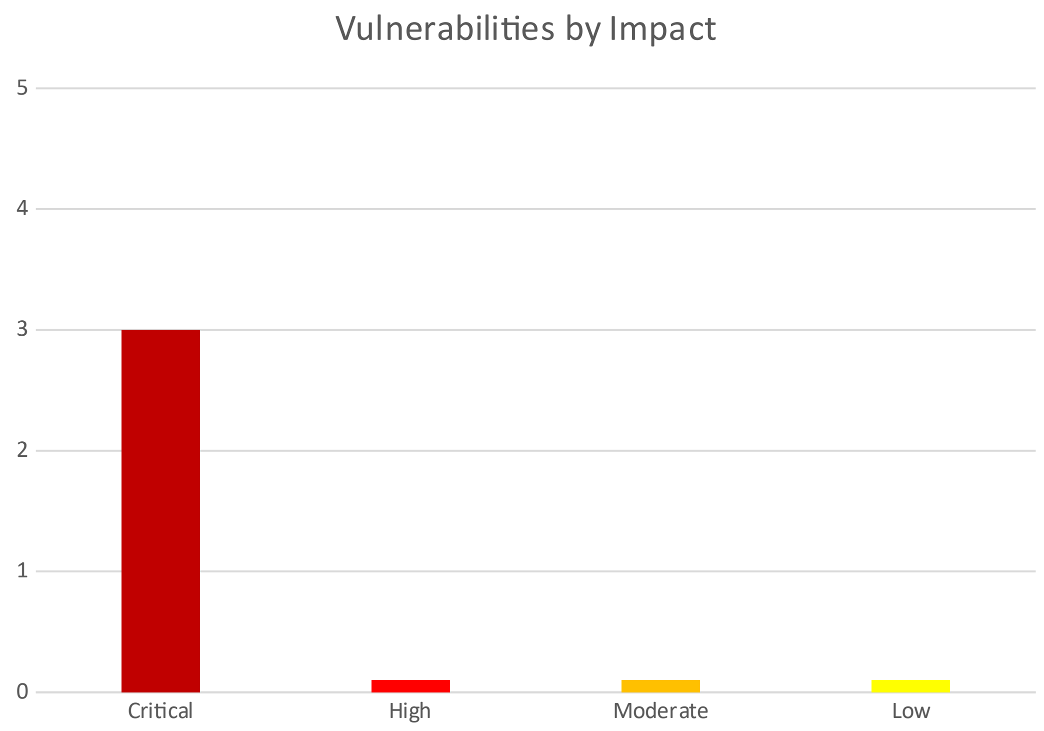
The extracted password hashes revealed a weak password policy. Jigsaw64S was able to crack these weak passwords, which granted access to the system. The use of predictable and easily guessable passwords significantly increased the risk of unauthorized access.

Client-Side Validation for the File Upload

Jigsaw64S identified that file upload functionality only performed validation on the client side, allowing for arbitrary file uploads. This weakness was exploited to upload a malicious reverse shell, leading to remote access to the system.

Vulnerabilities by Impact

The following chart illustrates the vulnerabilities found by impact:



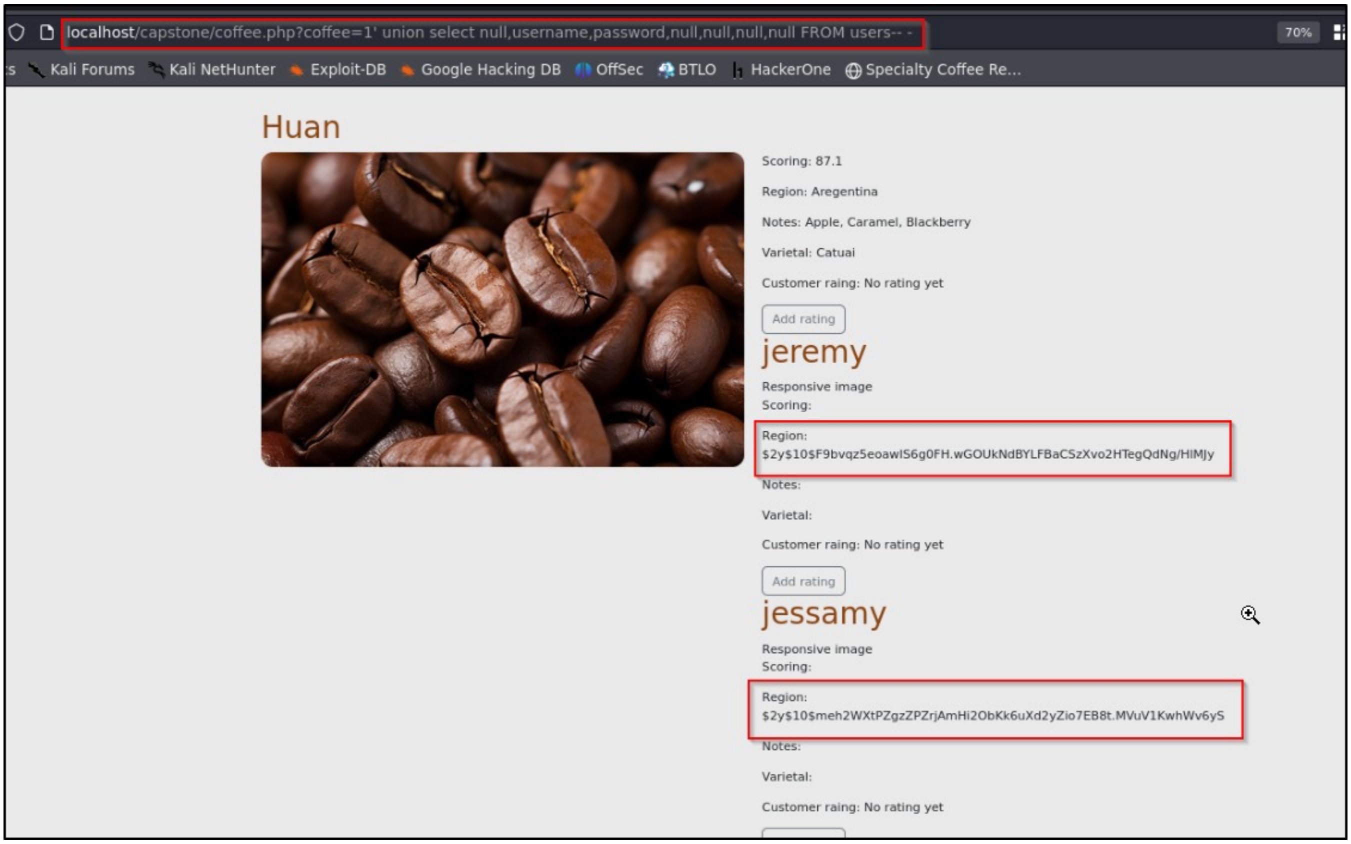
Web Application Penetration Test Findings

SQL Injection Vulnerability – Web Application (Critical)

Description:	CC's web application was found to be vulnerable to SQL injection attacks. This critical vulnerability allowed attackers to execute arbitrary SQL commands on the database, leading to unauthorized access to sensitive data. Jigsaw64S exploited this vulnerability to gain unauthorized access to the system and retrieve password hashes.
Impact:	Critical
System:	N/A
References:	[OWASPTop Ten: Injection]

Exploitation Proof of Concept

Jigsaw64S identified and exploited an SQL injection vulnerability in CC's web application. By injecting malicious SQL commands into the login form, Jigsaw64S was able to retrieve password hashes from the database. The retrieved hashes were then cracked using a password-cracking tool which revealed weak passwords.



```
localhost/capstone/coffee.php?coffee=1' union select null,username,password,null,null,null,null FROM users-- -
```

Huan

Scoring: 87.1

Region: Aregentina

Notes: Apple, Caramel, Blackberry

Varietal: Catuai

Customer raing: No rating yet

jeremy

Responsive image
Scoring:

Region:
$2y$10$F9bvqz5eoawIS6g0FH.wGOUkNdBYLFBaCSzXvo2HTegQdNg/HIMJy

Notes:

Varietal:

Customer raing: No rating yet

jessamy

Responsive image
Scoring:

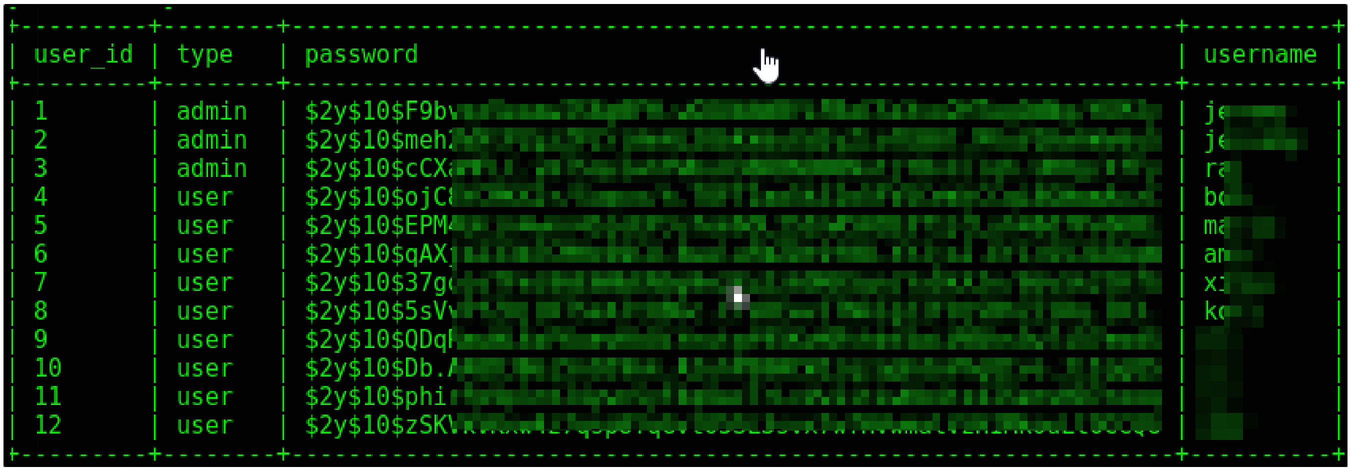
Region:
$2y$10$meh2WXtPZgzZPZrjAmHi2ObKk6uXd2yZio7EB8t.MVuV1KwhWv6yS

Notes:

Varietal:

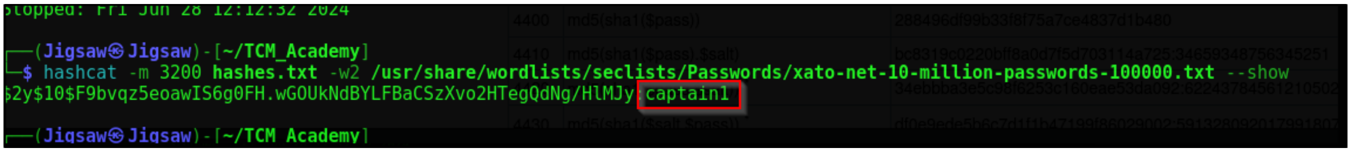
Customer raing: No rating yet

Figure 1: User password hashes discovered via SQL injection



```
+----------+--------+--------------------------------------------------------------+----------+
| user_id  | type   | password                                                     | username |
+----------+--------+--------------------------------------------------------------+----------+
| 1        | admin  | $2y$10$F9bv[redacted]                                        | je[redacted] |
| 2        | admin  | $2y$10$meh[redacted]                                         | je[redacted] |
| 3        | admin  | $2y$10$cCX[redacted]                                         | ra[redacted] |
| 4        | user   | $2y$10$ojC[redacted]                                         | bo[redacted] |
| 5        | user   | $2y$10$EPM[redacted]                                         | ma[redacted] |
| 6        | user   | $2y$10$qAX[redacted]                                         | am[redacted] |
| 7        | user   | $2y$10$37g[redacted]                                         | xi[redacted] |
| 8        | user   | $2y$10$5sV[redacted]                                         | ko[redacted] |
| 9        | user   | $2y$10$QDq[redacted]                                         | [redacted] |
| 10       | user   | $2y$10$Db.[redacted]                                         | [redacted] |
| 11       | user   | $2y$10$phi[redacted]                                         | [redacted] |
| 12       | user   | $2y$10$zSK[redacted]                                         | [redacted] |
+----------+--------+--------------------------------------------------------------+----------+
```

Figure2: User password column dumped from user table



```
Stopped: Fri Jun 28 12:12:32 2024

┌──(Jigsaw㉿Jigsaw)-[~/TCM_Academy]
└─$ hashcat -m 3200 hashes.txt -w2 /usr/share/wordlists/seclists/Passwords/xato-net-10-million-passwords-100000.txt --show
$2y$10$F9bvqz5eoawIS6g0FH.wGOUkNdBYLFBaCSzXvo2HTegQdNg/HlMJy:captain1

┌──(Jigsaw㉿Jigsaw)-[~/TCM_Academy]
```

Figure 2: Cracked admin hash revealing weak password (Revealed since this is a fictitious company)

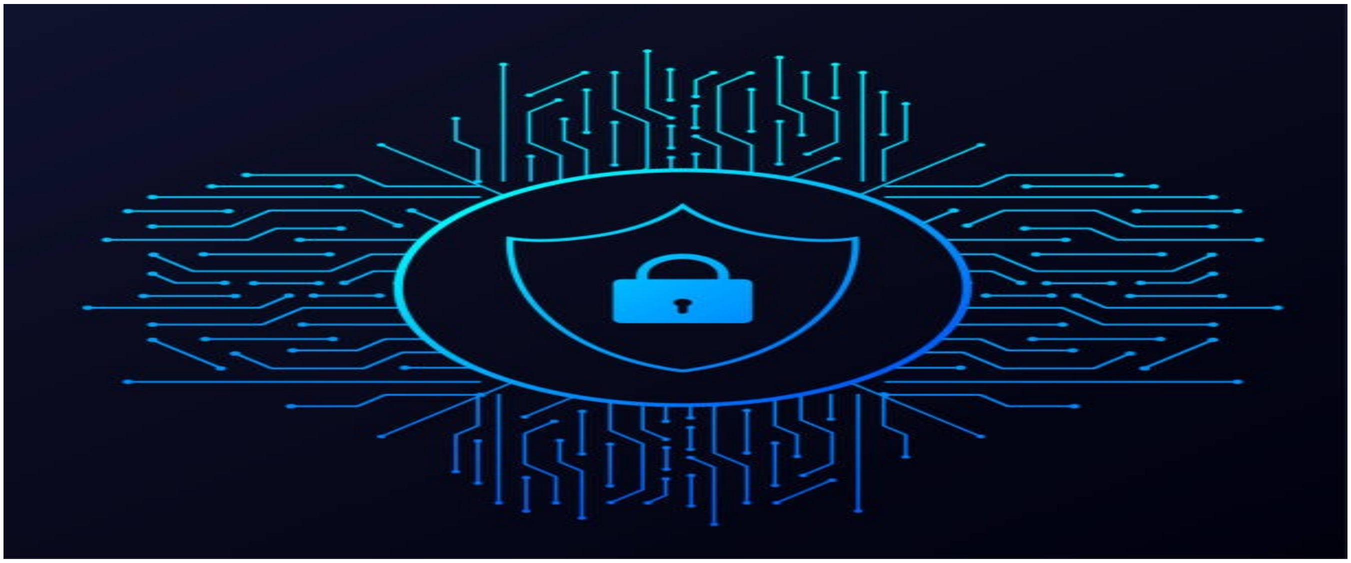
Remediation

Who:	IT Team
Vector:	Remote
Action:	**Item 1: Cross-Site Scripting (XSS) Vulnerability** CC should sanitize and validate all user inputs and outputs to prevent XSS attacks. Implement output encoding and input validation to mitigate XSS risks effectively. **Item 2: SQL Injection Vulnerability** Jigsaw64S recommends CC implement parameterized queries and prepared statements to mitigate SQL injection vulnerabilities. Additionally, regular code reviews and automated testing for SQL injection should be conducted to ensure the application's security. **Item 3: Weak Password Policy** Jigsaw64S recommends CC enforce a robust password policy, per the Center for Internet Security (CIS), including: 14 characters or longer Use different passwords for each account accessed Do not use words and proper names in passwords, regardless of language Item 3: Arbitrary File Upload Validation CC should implement server-side validation for file uploads to prevent arbitrary file uploads. This includes checking file types, sizes, and contents to ensure only legitimate files are accepted. **Item 4: Arbitrary File Upload Validation** CC should implement server-side validation for file uploads to prevent arbitrary file uploads. This includes checking file types, sizes, and contents to ensure only legitimate files are accepted. **Item 5: Multi-Factor Authentication (MFA)** Jigsaw64S recommends CC implement and enforce MFA across all web application-facing login services to add an additional layer of security.

Additional Reports and Scans (Informational)

Jigsaw64S provides all clients with all report information gathered during testing. This includes vulnerability scans and a detailed findings spreadsheet. For more information, please see the following documents:

- **Coffee Corp-867-19 Full Findings.xslx**
- **Coffee Corp-867-19 Vulnerability Scan Summary.xslx**
- **Coffee Corp-867-19 Vulnerability Scan by Host.pdf**



Last Page